

Risk-Sensitive Inverse Reinforcement Learning via Gradient Methods

Lillian J. Ratliff

Department of Electrical Engineering
University of Washington
Seattle, WA 98185
ratliff1@uw.edu

Eric Mazumdar

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720
mazumdar@berkeley.edu

Abstract

We address the problem of *inverse reinforcement learning* in Markov decision processes where the agent is *risk-sensitive*. In particular, we model risk-sensitivity in a reinforcement learning framework by making use of models of human decision-making having their origins in behavioral psychology, behavioral economics, and neuroscience. We propose a gradient-based inverse reinforcement learning algorithm that minimizes a loss function defined on the observed behavior. We demonstrate the performance of the proposed technique on two examples, the first of which is the canonical *Grid World* example and the second of which is a MDP modeling passengers' decisions regarding ride-sharing. In the latter, we use pricing and travel time data from a ride-sharing company to construct the transition probabilities and rewards of the MDP.

1 Introduction

The modeling and learning of human behaviors is becoming increasingly important as critical systems begin to rely more on automation and artificial intelligence. Yet, in this task we face a number of challenges, not least of which is the fact that humans are known to behave in ways that are not completely rational. For example, there is mounting evidence to support the fact that humans often use *reference points*—experiences that are perceived to be related to the decision the human is making. It has also been observed that their decisions are impacted by their perception of the external world (exogenous factors) and their present state of mind (endogenous factors) as well as how the decision is *framed* or presented [1, 2].

The success of *descriptive* behavioral models in capturing human behavior has long been touted by the psychology community and, more recently, by the economics community. In the engineering context, humans have largely been modeled, under rationality assumptions, from the so-called *normative* point of view where things are modeled *as they ought to be*, which is counter to a descriptive *as is* point of view. However, risk-sensitivity in the context of learning to control stochastic dynamic systems (see, e.g., [3–5]) has been fairly extensively explored in engineering and computer science. Many of these approaches are targeted at mitigating risks due to uncertainties in controlling a system such as a plant or robot. Much of this work simply handles *risk-aversion* by leveraging techniques such as exponential utility functions or minimizing variance.

On the other hand, human decision makers can be at once risk-averse and risk-seeking depending their frame of reference. Complex risk-sensitive behavior arising from human interaction with automation is only recently coming into focus. The adoption of diverse behavioral models in engineering—in particular, in learning and control—is growing due to the fact that humans are increasingly playing an integral role in automation both at the individual and societal scale. Learning accurate models of human decision-making is important for both *prediction* and *description*. For example, control/incentive schemes need to predict human behavior as a function of external stimuli. On the other hand, policy makers, *e.g.*, are interested in interpreting human reactions to implemented regulations and policies.

A few approaches for integrating risk-sensitivity of humans in the control and reinforcement learning problems via behavioral models [6–9] have recently emerged. These approaches largely assume a risk-sensitive Markov decision process (MDP) formulated based on a behavioral model and determine the optimal policy via, *e.g.*, a learning procedure. We refer to this as the *forward* problem. We are interested in solving the so-called *inverse* problem which seeks to estimate the decision-making model given a set of demonstrations. In particular, we may care about producing the reward function (or at least, characterize the space of reward functions) that results in similar behavior. On the other hand, we may want to extract the optimal policy from a set of demonstrations so that we can reproduce the behavior in support of, *e.g.*, designing incentives or control policies.

In particular, in this paper we model human decision-makers as *risk-sensitive Q-learning agents* where we exploit very rich behavioral models from behavioral psychology and economics as well as neuroscience that capture a whole spectrum of risk-sensitivity. We propose a gradient-based learning algorithm for inferring the decision-making model parameters from demonstrations—that is, we propose a novel framework for solving the *risk-sensitive inverse reinforcement learning* problem. We demonstrate the efficacy of the learning scheme on i) the canonical Grid World example and ii) a passenger’s view of ride-sharing as an MDP with model parameters estimated from real-world data.

The remainder of this paper is organized as follows. In Section 2, we briefly overview the Q-learning model we assume for risk-sensitive agents and show that it is amenable to integration with the behavioral models. In Section 3, we formulate the inverse reinforcement learning problem and propose a gradient-based algorithm to solve it. Examples that demonstrate the ability of the proposed scheme to capture a wide breadth of risk-sensitive behaviors are provided in Section 4. Finally, we conclude with some discussion and comments on future work in Section 5.

2 Risk-Sensitive Reinforcement Learning

In order to learn a decision-making model for an agent who faces sequential decisions in an uncertain environment, we adopt a recently developed risk-sensitive reinforcement learning framework that allows for integration of behavioral decision-making models [8, 7, 6]. Under the assumption that the agent is making decisions according to this model, we then formulate a gradient-based method for learning the policy as well as parameters of the agent’s value function.

Risk-sensitivity has been long covered in the stochastic decision-making literature. The typical way in which risk is dealt with is through transformations of the return, *e.g.*, by an exponential value function. As an alternative approach, transformation of the *temporal differences* in a Q-learning scheme according to a risk model has recently been proposed [7–9]. Transforming of the temporal differences avoids certain pitfalls of the reward transformation approach such as poor convergence performance. We adopt this idea in order to generate *forward* models of agent decision-making that are able to capture diverse behavior profiles which we then use in our *inverse* learning algorithm to estimate their behavior.

2.1 Markov Decision Process

We will consider a class of finite MDPs which consist of a state space X , an admissible action space $A(x) \subset A$ for each $x \in X$, a transition kernel $P(x'|x, a)$ (which denotes the probability of moving from state x to x' given action a), and a reward function $r : X \times A \times W \rightarrow \mathbb{R}$ where W is the disturbance space and has distribution $P_r(\cdot|x, a)$. Including disturbances w allows us to model random rewards for which use the notation $R(x, a)$ to denote the random reward having distribution $P_r(\cdot|x, a)$.

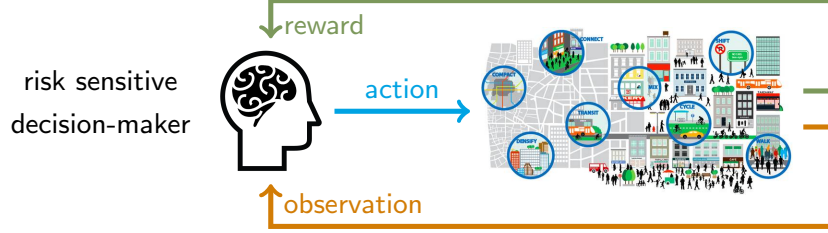


Figure 1: Human decision-makers are often not completely rational and their decisions are impacted by various exogenous and endogenous factors including risk. We model them as risk-sensitive agents attempting to solve a MDP via risk-sensitive reinforcement learning.

In the classical expected utility maximization framework, the agent seeks to maximize the expected discounted rewards by selecting a Markov policy π . That is, for an infinite horizon MDP, the optimal policy is obtained by maximizing

$$J(x_0) = \max_{\pi} \mathbb{E} \left[\sum_{t=1}^{\infty} \gamma^t R(x_t, a_t) \right] \quad (1)$$

where x_0 is the initial state and $\gamma \in [0, 1)$ is the discount factor. The risk-sensitive reinforcement learning problem transforms the above problem to account for a salient features of human decision-making such as loss aversion, reference point dependence, and framing effects.

2.2 Value Functions

Given the environmental uncertainty and uncertainty in the reward, we model the outcome of each action as a real-valued random variable $Y(i) \in \mathbb{R}$, $i \in I$ where I denotes a finite event space and Y is the outcome of i -th event with probability $\mu(i)$, $\mu \in \mathcal{P}$, the space of probability distributions on I . Much like the standard expected utility framework, an agent makes choices based on the value of their outcome as defined by their *value function* $u : \mathbb{R} \rightarrow \mathbb{R}$.

There are a number of existing approaches to defining value functions that capture risk-sensitivity and loss aversion. These approaches derive from a variety of fields including behavioral psychology/economics, mathematical finance, and even neuroscience.

One of the most salient features of human decision-making is that losses are perceived more significant than a gain of equal true value. The models with the greatest efficacy in capturing this are both convex and concave in different regions of the outcome space. Prospect theory, developed by Kahneman and Tversky [10, 11], is built on one such model. The form of the value function introduced in prospect theory is given by

$$u(y) = \begin{cases} c_+(y - \bar{y})^{\rho_+}, & y > \bar{y} \\ -c_-(\bar{y} - y)^{\rho_-}, & y \leq \bar{y} \end{cases} \quad (2)$$

where \bar{y} is the reference point that the decision-maker compares outcomes against in determining if the decision is a loss or gain. The parameters $(c_+, c_-, \rho_+, \rho_-)$ control the degree of risk-sensitivity and loss-aversion. For example, $0 < \rho_+, \rho_- < 1$ leads to risk-averse preferences on gains and risk-seeking preferences on losses (concave in gains, convex in losses); $\rho_+ = \rho_- = 1$ leads to risk-neutral preferences; $\rho_+, \rho_- > 1$ leads to risk-averse preferences on losses and risk-seeking preferences on gains (convex in gains, concave in losses). Prospect-theoretic risk preferences such as framing effects, reference dependence and loss aversion have been widely observed in experimental studies on human decision-making (see, *e.g.*, [12–14]). Experimental results for a series of one-off decisions have indicated that typically both ρ_+ and ρ_- are less than one thereby indicating that humans are risk-averse on gains and risk-seeking on losses—that is, u is concave for $y > \bar{y}$ and convex otherwise).

In addition to the non-linear transformation of outcome values, in prospect theory the effect of under/over-weighting the likelihood of events that has been commonly observed in human behavior is model via *warping* of event probabilities [15, 16].

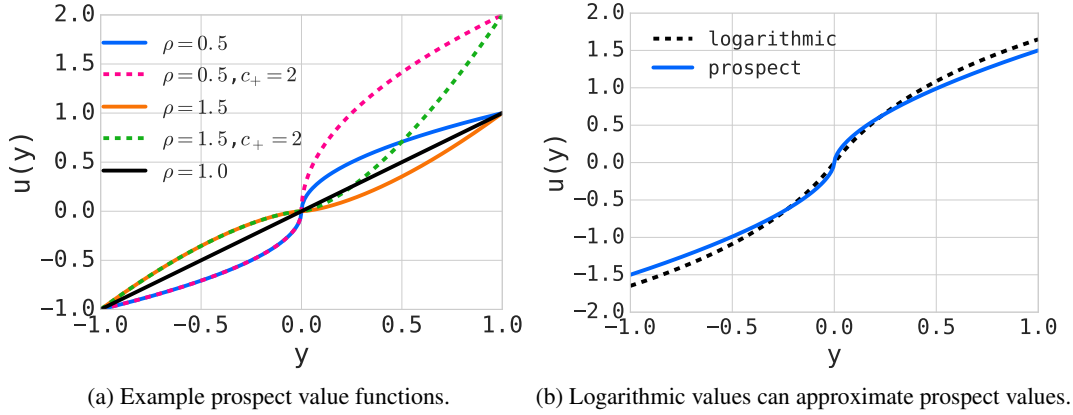


Figure 2: Value functions with different parameters.

Outside of the prospect theory value, other mappings have been proposed to capture risk-sensitivity. One example is the entropic map $u(y) = \exp(\lambda y)$, where λ controls the degree of risk-sensitivity. Another is the linear map defined by

$$u(y) = \begin{cases} (1 - \kappa)y, & y > 0 \\ (1 + \kappa)y, & y \leq 0 \end{cases} \quad (3)$$

which was proposed in [8] where they formulate a risk-sensitive reinforcement learning procedure. Here, $\kappa \in (-1, 1)$ controls the degree of risk sensitivity.

We also introduce a logarithm-based value function and an exponential-based value function both of which approximate the shape of the prospect theory value—*i.e.* convex in gains and concave in losses—while improving the performance (in terms of convergence speed) of the gradient-based inverse reinforcement learning algorithm we propose in Section 3. In particular, we consider the logarithmic value function given by

$$u(y) = \begin{cases} c_+ \log(1 + \rho_+(y - \bar{y})), & y > \bar{y} \\ -c_- \log(1 + \rho_-(\bar{y} - y)), & y \leq \bar{y} \end{cases} \quad (4)$$

and the exponential value function give by

$$u(y) = \begin{cases} \frac{1}{1 + \exp(-\lambda_+(y - \bar{y}))} - \frac{1}{1 + \exp(\lambda_+(y - \bar{y}))}, & y > \bar{y} \\ \frac{1}{1 + \exp(\lambda_-(\bar{y} - y))} - \frac{1}{1 + \exp(-\lambda_-(\bar{y} - y))}, & y \leq \bar{y} \end{cases} \quad (5)$$

where $\lambda_+ < 0$ and $\lambda_- > 0$. The logarithmic value function is Lipschitz and hence, the theoretical results we present in Section 3.2 apply. The exponential value function, on the other hand, has only two parameters and is therefore easier to learn. Unlike the entropic map, it allows for a convex-concave structure in losses and gains.

It is our aim to bring these models and ideas to a risk-sensitive inverse reinforcement learning setting and design algorithms for estimation which will lay the ground work for ultimately designing incentives as well as control policies.

2.3 Valuation Functions

To capture risk-sensitivity, *valuation functions* generalize the expectation operator which considers *average* or *expected* outcomes¹.

Definition 1 ([17, 7, 18]) A mapping $\mathcal{V} : \mathbb{R}^{|I|} \times \mathcal{P} \rightarrow \mathbb{R}$ is called a valuation function if for each $\mu \in \mathcal{P}$, (i) $\mathcal{V}(Y, \mu) \leq \mathcal{V}(Z, \mu)$ whenever $Y \leq Z$ (monotonic) and (ii) $\mathcal{V}(Y + y\mathbf{1}, \mu) = \mathcal{V}(Y, \mu) + y$ for any $y \in \mathbb{R}$ (translation invariant).

¹In the case of two events, the valuation function can also capture warping of probabilities. Alternative approaches to reinforcement learning based on cumulative prospect theory for the more general case have been examined [6]

Such a map is used to characterize an agent's preferences—that is, one prefers (Y, μ) to (Z, ν) whenever $\mathcal{V}(Z, \nu) \leq \mathcal{V}(Y, \mu)$. Let us consider the following example.

Example 1 (shortfall valuation) *It is common for human decision-makers to compare outcomes to reference points, \bar{y} , as we mentioned above. In addition, people may form an acceptance level that they compare their values of their value function for different outcomes too. Let's denote by \bar{r} the acceptance level and we assume that \bar{r} is in the range of the value function u . For prospect theoretic value functions, one reasonable value for \bar{r} is $u(\bar{y})$.*

The shortfall valuation defines a valuation function that incorporates reference and acceptance level dependence into the decision-making model. It originated in mathematical finance [19, 17] where it was shown to be a valuation function and it has been used in a risk-sensitive MDP context in [3] and in a reinforcement learning context in [7]. The shortfall \mathcal{V} induced by a value function $u : \mathbb{R} \rightarrow \mathbb{R}$ and an acceptance level \bar{r} is given by

$$\mathcal{V}(Y) = \sup \{m \in \mathbb{R} \mid \mathbb{E}_Y[u(Y - m)] \geq \bar{r}\} \quad (6)$$

where we suppress the dependence of \mathcal{V} on μ , the distribution of the random variable for outcomes Y . ■

Now for each state–action pair, we can define a valuation function. Let us then call $\mathcal{V}(Y|x, a) : \mathbb{R}^{|I|} \times X \times A \rightarrow \mathbb{R}$ a valuation map such that $\mathcal{V}_{x,a} \equiv \mathcal{V}(\cdot|x, a)$ is a valuation function. The following is an example of a state-action dependent valuation function.

Example 2 (entropic valuation) *The entropic map, introduced above, has been used extensively in the field of risk measures [17], in neuroscience to capture risk sensitivity in motor control [9] and even more so in control of MDPs (see, e.g., [20, 21, 4]). The entropic risk valuation is defined by*

$$\mathcal{V}(Y) = \frac{1}{\lambda} \log \mathbb{E}[\exp(\lambda Y)] \quad (7)$$

Using the shortfall valuation with acceptance level \bar{r} [17], we have

$$\mathcal{V}_{x,a}(Y) = \sup \{m \in \mathbb{R} \mid \mathbb{E}_{x,a}[\exp(\lambda(Y - m))] \geq r_0\} = \frac{1}{\lambda} \log \mathbb{E}_{x,a}[\exp(\lambda Y)] - \frac{1}{\lambda} \log(\bar{r}). \quad (8)$$

With $\bar{r} = 1$, it has Taylor series expansion $\mathcal{V}_{x,a}(Y) = \mathbb{E}[Y] + \lambda \text{Var}[Y] + O(\lambda^2)$ so that the parameter $\lambda \in \mathbb{R}$ controls the risk preference of \mathcal{V} —that is, if $\lambda > 0$, \mathcal{V} is everywhere convex and thus expresses risk-aversion whereas if $\lambda < 0$, \mathcal{V} is everywhere concave and thus expresses risk-seeking. ■

If we let $\mathcal{V}_s^\pi(Y) = \sum_{a \in A(x)} \pi(a|x) \mathcal{V}_{x,a}(Y)$, the optimization problem in (1) generalizes to

$$\tilde{J}_T(\pi, x_0) = \mathcal{V}_{x_0}^{\pi_0} \left[R[x_0, a_0] + \gamma \mathcal{V}_{x_1}^{\pi_1} [R[x_1, a_1] + \cdots + \gamma \mathcal{V}_{x_T}^{\pi_T} [R(x_T, a_T)] \cdots] \right]. \quad (9)$$

Let us define $\max_\pi \tilde{J}(\pi, x_0) = \lim_{T \rightarrow \infty} \tilde{J}_T(\pi, x_0)$.

2.4 Risk-Sensitive Q-Learning Convergence

In the classical reinforcement learning framework, the Bellman equation is used to derive a Q-learning procedure. Generalization of the Bellman equation for risk-sensitive reinforcement learning have been derived in [3, 8]. These generalizations are then used to formulate an action–value function or Q-learning procedure for the risk-sensitive reinforcement learning problem. In particular, as shown in [3], if V^* satisfies

$$V^*(x_0) = \max_{a \in A(x)} \mathcal{V}_{x,a}(R(x, a) + \gamma V^*), \quad (10)$$

then $V^* = \max_\pi \tilde{J}(\pi, x_0)$ holds for all $x_0 \in X$; moreover, a deterministic policy is optimal if $\pi^*(x) = \arg \max_{a \in A(x)} \mathcal{V}_{x,a}(R + \gamma V^*)$ [3, Thm. 5.5]. The action–value function $Q^*(x, a) = \mathcal{V}_{x,a}(R + \gamma V^*)$ is defined such that (10) becomes

$$Q^*(x, a) = \mathcal{V}_{x,a} (R + \gamma \max_{a' \in A(x')} Q^*(x', a')), \quad (11)$$

for all $(x, a) \in X \times A$.

The shortfall valuation function as in [7] allows us to consider a range of non-trivial reference points. Recalling (6), we consider the valuation function $\mathcal{V}_{s,a}(Y) = \sup \{z \in \mathbb{R} \mid \mathbb{E}[u(Y - z)] \geq \bar{r}\}$ where the

expectation is taken with respect to $\mu = P(x'|x, a)P_r(w|x, a)$. Hence, if u is continuous and strictly increasing, then $\mathcal{V}_{x,a}(Y) = z^*(x, a)$ is the unique solution to $\mathbb{E}[u(Y - z^*(x, a))] = \bar{r}$ (see [17, Prop. 4.104] or [7, Prop. 3.1]).

With $Y = R + \gamma V^*$, $z^*(x, a)$ corresponds to $Q^*(x, a)$ and, in particular,

$$\mathbb{E}[u(r(x, a, w) + \gamma \max_{a' \in A(x')} Q^*(x', a') - Q^*(x, a))] = \bar{r} \quad (12)$$

where, again, the expectation is taken with respect to $\mu = P(x'|x, a)P_r(w|x, a)$.

The above leads naturally to a Q-learning procedure,

$$Q(x_t, a_t) \leftarrow Q(x_t, a_t) + \alpha_t(x_t, a_t)[u(r_t + \gamma \max_a Q(x_{t+1}, a) - Q(x_t, a_t)) - \bar{r}], \quad (13)$$

where we would like to point out that the mapping u is applied to the temporal difference $r_t + \gamma \max_a Q(x_{t+1}, a) - Q(x_t, a_t)$ instead of simply the reward r_t . This procedure has convergence guarantees even in this more general setting under some assumptions on the value function u .

Theorem 1 (Q-learning Convergence [7, Thm. 3.2]) *Suppose that $u : Y \rightarrow \mathbb{R}$ is in $C(Y, \mathbb{R})$, is strictly increasing in y and there exists constants $\varepsilon, L > 0$ such that $\varepsilon \leq \frac{u(y) - u(y')}{y - y'} \leq L$ for all $y \neq y'$. Moreover, suppose that there exists a y such that $u(y) = \bar{r}$. If the non-negative learning rates $\alpha_t(x, a)$ are such that $\sum_{t=0}^{\infty} \alpha_t(x, a) = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2(x, a) < \infty$, $\forall (x, a) \in X \times A$, then the procedure in (13) converges to $Q^*(x, a)$ for all $(x, a) \in X \times A$ with probability one.*

The assumptions on α_t are fairly standard and the core of the convergence proof is based on the Robbins and Siegmund Theorem [22].

3 Risk-Sensitive Inverse Reinforcement Learning

We formulate the risk-sensitive inverse reinforcement learning problem in a general form as follows. First, we select a parametric class of policies, $\{\pi_\theta\}_\theta$, $\pi_\theta \in \Pi$ and parametric value function $\{u_\theta\}_\theta$, $u_\theta \in \mathcal{F}$ where \mathcal{F} is a family of value functions and $\theta \in \Theta \subset \mathbb{R}^d$.

We will use value functions such as those described in Section 2.2. For example, if u is the prospect theory value function defined in (2), then the parameter vector is $\theta = (c_+, c_-, \rho_+, \rho_-, \gamma, \beta)$. For mappings u and Q , we will now indicate their dependence on θ —that is, we will write $Q(x, a, \theta)$ and $u_\theta(y) = u(y, \theta)$ where $u : Y \times \Theta \rightarrow \mathbb{R}$. Note that since y is the temporal difference it also depends on θ and we will indicate that where not obvious by writing $y(\theta)$.

It is common in the inverse reinforcement learning literature to adopt a smooth map G that operates on the action-value function space for defining the parametric policy space—e.g., Boltzmann policies of the form

$$G_\theta(Q)(a|x) = \frac{\exp(\beta Q(x, a, \theta))}{\sum_{a' \in A} \exp(\beta Q(x, a', \theta))} \quad (14)$$

to the action-value functions Q where $\beta > 0$ controls how close $G_\theta(Q)$ is to a *greedy policy* which we define to be any policy π such that $\sum_{a \in A} \pi(a|x)Q(x, a, \theta) = \max_{a \in A} Q(x, a, \theta)$ —in this case, we say π is greedy with respect to $Q(\cdot, \cdot, \theta)$. We will utilize policies of this form. Note that, as is pointed out in [23], the benefit of selecting strictly stochastic policies is that if the true agent's (expert's) policy is deterministic, uniqueness of the solution is forced.

We aim to *tune* the parameters so as to minimize some loss $\ell(\pi_\theta)$ which is a function of the parameterized policy π_θ . We will use the shorthand notation $\ell(\theta) = \ell(\pi_\theta)$.

3.1 Inverse Reinforcement Learning Optimization Problem

The optimization can be stated generally as

$$\min_{\theta \in \Theta} \{\ell(\theta) \mid \pi_\theta = G_\theta(Q^*), u_\theta \in \mathcal{F}\} \quad (15)$$

Given a set of *demonstrations* $\mathcal{D} = \{(x_k, a_k)\}_{k=1}^N$, it is our goal to recover the policy and estimate the value function. There are several possible loss functions. For example, suppose we elect to

minimize the negative weighted log-likelihood of the demonstrated behavior which is given by

$$\ell(\theta) = \sum_{(x,a) \in \mathcal{D}} w(x,a) \log(\pi_\theta(x,a)) \quad (16)$$

where $w(x,a)$ may, *e.g.*, be the normalized empirical frequency of observing (x,a) pairs in \mathcal{D} , i.e. $n(x,a)/N$ where $n(x,a)$ is the frequency of (x,a) . Related to maximizing the log-likelihood, an alternative loss function may be formulated as the relative entropy or Kullback–Leibler (KL) divergence between the empirical distribution of the state-action trajectories and their distribution under the learned policy—that is, we could minimize

$$\ell(\theta) = \sum_{x \in \mathcal{D}_x} D_{\text{KL}}(\hat{\pi}(\cdot|x) || \pi_\theta(\cdot|x)) \quad (17)$$

where $D_{\text{KL}}(P||Q) = \sum_i P(i) \log(P(i)/Q(i))$, $\mathcal{D}_x \subset \mathcal{D}$ is the sequence of observed states and $\hat{\pi}$ is the empirical distribution on the trajectories of \mathcal{D} .

3.2 Gradient-Based Risk-Sensitive Inverse Reinforcement Learning

We propose to solve the problem of estimating the parameters of the agent’s value function and approximating the agent’s policy via gradient methods which requires computing the derivative of $Q^*(x,a,\theta)$ with respect to θ . Hence, given the form of the Q-learning procedure where the temporal differences are transformed as in (13), we need to derive a mechanism for obtaining the optimal Q , show that it is in fact differentiable, and derive a procedure for obtaining the derivative.

Given the form of smoothing map G_θ given in (14), we can compute the derivative of the policy π_θ with respect to θ_k , an element of Θ :

$$D_{\theta_k} \pi_\theta(a|x) = \pi_\theta(a|x) D_{\theta_k} \ln(\pi_\theta(a|x)) \quad (18)$$

$$= \pi_\theta(a|x) \beta \left(D_{\theta_k} Q^*(x,a,\theta) - \sum_{a' \in A} \pi_\theta(a'|x) D_{\theta_k} Q^*(x,a',\theta) \right) \quad (19)$$

We will show that $D_{\theta_k} Q_\theta^*$ can be calculated almost everywhere on Θ by solving fixed-point equations similar to the Bellman-optimality equations.

We will require some assumptions on u .

Assumption 1 *The value function $u : Y \times \Theta \rightarrow \mathbb{R}$ satisfies the following: (i) it is strictly increasing in y and for each $\theta \in \Theta$, there exists a \bar{r} such that $u(\bar{y}, \theta) = \bar{r}$; (ii) for each $\theta \in \Theta$, it is Lipschitz in y with constant $L_y(\theta)$ and locally Lipschitz on Θ with constant L_θ ; (iii) there exists $\varepsilon > 0$ such that $\varepsilon \leq \frac{u(y,\theta) - u(y',\theta)}{y - y'}$; and (iv) $u \in C^1(Y \times \Theta, \mathbb{R})$.*

Define $L_y = \min_\theta L_y(\theta)$ and $L = \max_\theta \{L_y(\theta), L_\theta\}$. Let $\tilde{u} \equiv u - \bar{r}$. Let us re-write our Q -update equation in the following form (one that is standard, *e.g.*, in the stochastic approximation algorithm literature [24–26]):

$$Q_{t+1}(x,a,\theta) = \left(1 - \frac{\alpha_t}{\alpha}\right) Q_t(x,a,\theta) + \frac{\alpha_t}{\alpha} (\alpha(u(y_t(\theta), \theta) - \bar{r}) + Q_t(x,a,\theta)) \quad (20)$$

where $\alpha \in (0, \min\{L^{-1}, 1\})$ and we have suppressed the dependence of α_t on (x,a) . Then we define the map T such that

$$(TQ)(x,a,\theta) = \alpha \mathbb{E} \tilde{u}(y(\theta), \theta) + Q(x,a,\theta) \quad (21)$$

where $y(\theta) = r(x,a,w) + \gamma \max_{a' \in A} Q(x',a,\theta) - Q(x,a,\theta)$.

This mapping is a contraction: for cases where $\bar{r} = 0$, it was first shown in [8] and in the more general setting using similar techniques in [7].

Theorem 2 *Assume that $u : Y \times \Theta \rightarrow \mathbb{R}$ satisfies Assumption 1. Then the following statements hold:*

- (a) Q_θ^* is locally Lipschitz-continuous as a function of θ —that is, for any $(x,a) \in X \times A$, $\theta, \theta' \in \Theta$, $|Q^*(x,a,\theta) - Q^*(x,a,\theta')| \leq R \|\theta - \theta'\|$ for some $R > 0$;

(b) Except on a set of measure zero, the gradient $D_\theta Q_\theta^*$ is given by the solution of the fixed-point equation

$$\begin{aligned} \phi_\theta(x, a) = & \alpha \mathbb{E}^\mu \left[D_\theta \tilde{u}(y(\theta), \theta) + D_y \tilde{u}(y(\theta), \theta) \cdot \left(\gamma \sum_{a' \in A} \pi(a'|x) \phi_\theta(x', a') - \phi_\theta(x, a) \right) \right] \\ & + \phi_\theta(x, a) \end{aligned} \quad (22)$$

where π is any policy greedy with respect to $Q(\cdot, \cdot, \theta)$.

We provide the proof in Appendix A. To give a high-level outline, we use an induction argument combined with a contraction mapping argument on the map

$$\begin{aligned} (S\phi_\theta)(x, a) = & \alpha \mathbb{E}^\mu \left[D_\theta \tilde{u}(y(\theta), \theta) + D_y \tilde{u}(y(\theta), \theta) \left(\gamma \sum_{a' \in A} \pi(a'|x) \phi_\theta(x', a') - \phi_\theta(x, a) \right) \right] \\ & + \phi_\theta(x, a) \end{aligned} \quad (23)$$

where \mathbb{E}^μ is the conditional expectation with respect to $\mu = P(x'|x, a)P_r(w|x, a)$ to show part (a). The almost everywhere differentiability follows from Rademacher's Theorem (see, e.g., [27, Thm. 3.1]).

Theorem 2 gives us a procedure—namely, a fixed-point equation which is a contraction—to compute the derivative $D_{\theta_k} Q^*$ so that we can compute the derivative of our loss function $\ell(\theta)$. Hence the gradient method provided in Algorithm 1 for solving risk-sensitive inverse reinforcement learning problem is well formulated.

Algorithm 1 Gradient-Based Risk-Sensitive Inverse Reinforcement Learning

```

1: procedure RISKIRL( $\mathcal{D}$ )
2:   Initialize:  $\theta \leftarrow \theta_0$ 
3:   while  $k < \text{MAXITER}$  &  $\|\ell(\theta) - \ell(\theta_-)\| \geq \delta$  do
4:      $\theta_- \leftarrow \theta$ 
5:      $\eta_k \leftarrow \text{LINESEARCH}(\ell(\theta_-), D_\theta \ell(\theta_-))$ 
6:      $\theta \leftarrow \theta_- - \eta_k D_\theta \ell(\theta_-)$ 
7:      $k \leftarrow k + 1$ 
8:   return  $\theta$ 

```

Remark 1 The prospect theory u given in (2) is not globally Lipschitz in y —in particular, it is not Lipschitz near the reference point \bar{y} —for values of ρ_+ and ρ_- in less than one. Moreover, for certain parameter combinations, it may not even be differentiable. In [7], the authors propose some techniques such as truncation for handling the reinforcement learning problem numerically when u is not globally Lipschitz. This also motivates the use of the logarithm-based value function that well-approximate the prospect theory u and retain the convex-concave structure. In practice, for cases where u is not Lipschitz, in the gradient-based algorithm above we are computing and implementing elements of the sub-differential.

3.3 Complexity

To properly understand how well Algorithm 1 performs for different amounts of data—small dataset size is often a challenge in modeling sequential human decision-making—we analyze the case when the loss function, $\ell(\theta)$, is either the sum over states of the KL divergence between the policy under our learned value function and the the empirical policy of the agent, or the negative of the log-likelihood of the data. These are given by (17) and (16) respectively.

Maximizing the log-likelihood is equivalent to minimizing a weighted sum over states of the KL divergence between the empirical policy of the optimal agent and the policy under the learned value function. In particular, the weighted log-likelihood in (16) can be re-written as

$$\ell(\theta) = \sum_{x \in \mathcal{D}_x} w(x) D_{KL}(\pi_\theta(\cdot|x) \parallel \hat{\pi}_n(\cdot|x)) \quad (24)$$

by subtracting $w(x, a) \log(w(x, a)/w(x))$, $\forall(x, a)$ where $\mathcal{D}_x \subset \mathcal{D}$ is the set of observed states and $w(x)$ is the frequency of state x normalized by $|\mathcal{D}| = N$ defined above. Both cost functions are natural metrics for performance in that they minimize a *distance* between the optimal policy under the learned agent and empirical policy of the true agent. This approach has the added benefit that it is independent of θ and therefore will not be affected by scaling of the value functions [23].

Given that we have access only to the empirical policy $\hat{\pi}_n$ and not the true policy π , we construct bounds on the performance as measured by the total variation (TV) metric on discrete probability distributions which is given by

$$\delta(\pi_\theta(\cdot|x), \pi(\cdot|x)) = \frac{1}{2} \|\pi_\theta(\cdot|x) - \pi(\cdot|x)\|_1. \quad (25)$$

We can upper bound $\delta(\pi_\theta(\cdot|x), \pi(\cdot|x))$ by

$$\delta(\pi_\theta(\cdot|x), \pi(\cdot|x)) \leq \delta(\hat{\pi}_n(\cdot|x), \pi_\theta(\cdot|x)) + \delta(\hat{\pi}_n(\cdot|x), \pi(\cdot|x)) \quad (26)$$

where $\delta(\hat{\pi}_n(\cdot|x), \pi_\theta(\cdot|x))$ can be thought of as the training error and $\delta(\hat{\pi}_n(\cdot|x), \pi(\cdot|x))$ is the distance between the empirical policy and the true policy in state x . The weak law of large numbers guarantees that $\hat{\pi}_n(\cdot|x) \rightarrow \pi(\cdot|x)$ point-wise in probability. From this fact and the Dvoretzky Kiefer-Wolfowitz inequality (see, e.g., [28, 29]), one can derive the following bound:

$$\Pr(\|\pi(\cdot|x) - \hat{\pi}_n(\cdot|x)\|_1 > \epsilon) \leq 2|A|e^{-2n\epsilon^2/|A|^2}, \quad \epsilon > 0 \quad (27)$$

where n is the number of samples from the distribution $\pi(s)$ and $|A|$ is the cardinality of the action set. Combining this bound with (26), we get that, with probability $1 - \nu$,

$$\delta(\pi_\theta(\cdot|x), \pi(\cdot|x)) \leq |A| \left(\frac{2}{n} \log \frac{2|A|}{\nu} \right)^{1/2} + \delta(\hat{\pi}_n(\cdot|x), \pi_\theta(\cdot|x)) \quad (28)$$

Thus, if for large N with $n \geq N$, we have that $\delta(\hat{\pi}_n(\cdot|x), \pi_\theta(\cdot|x)) \leq \bar{\epsilon}$, $\bar{\epsilon} > 0$, then the left-hand side of the inequality is $O(n^{-1/2})$.

We note that this bound is for each individual state x . Thus, for states that are visited more frequently by the agent, we have better guarantees on how well the policy under the learned value function approximates the true policy. Further, this bound on the per-state level suggests ways of designing data collection schemes to better understand the agent's actions in less explored regions of the state space.

4 Examples

Let us now demonstrate the proposed methods performance on two examples. While we are able to formulate the risk-sensitive inverse reinforcement learning problem for parameter vectors θ that include γ and β , in the following examples we use $\gamma = 0.95$ and $\beta = 0.5$. The purpose of doing this is to explore the effects of changing the value function parameters on the resulting policy.

4.1 Prospecting in Grid World

We tested Algorithm 1 on data from agents operating on the canonical Grid World MDP example shown in Figure 3 where the agents all start in the blue box, aim to maximize their value function over an infinite horizon. In particular, each square represents a state, and the action space is $A = \{N, NE, E, SE, S, SW, W, NW\}$. Each action corresponds to a movement in the specified direction. With probability 0.93, the agent moves in the desired direction and with probability 0.01, they move in any of the other seven directions. To make the grid finite, any action taking the agent out of the grid has probability zero, and the other actions are re-weighted accordingly. The black and hatched green states are absorbing and give rewards of -1 and $+1$, respectively. All other states give rewards of -0.1 .

For each of the agents, we use the shortfall valuation function with acceptance level $\bar{r} = 0$. For the prospect theory agents, we used reference point $\bar{y} = 0$. We selected these values in order to isolate the effects of the reference point and acceptance level from those of the value function parameters.

Figure 4 provides a visualization of the agent's risk sensitivity resulting from different parameter combinations using the logarithm-based value function for the true and learned agents. There are four

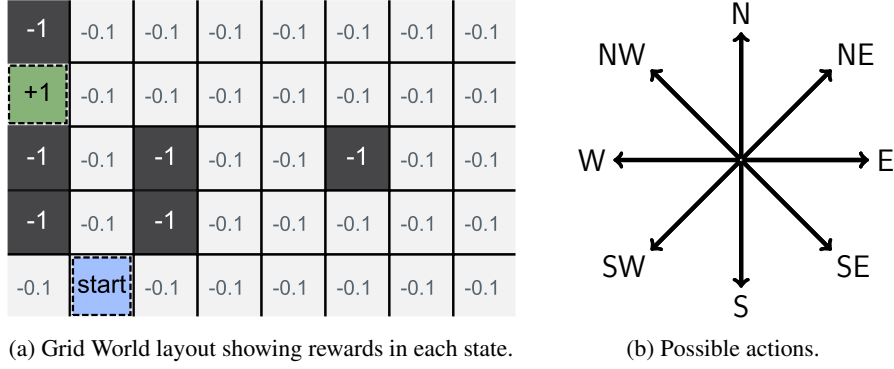


Figure 3: Canonical Grid World Example.

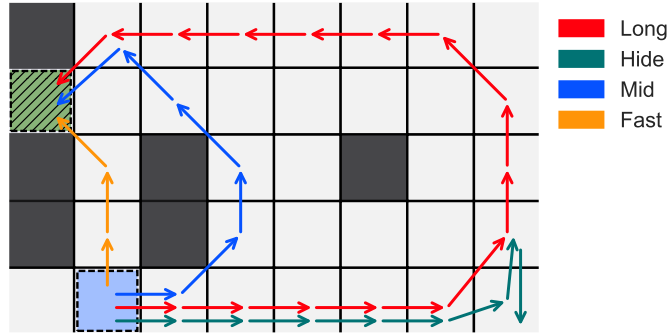


Figure 4: True and learned paths extracted from the optimal policies for agent's with logarithm-based value functions. The true and learned paths align exactly with one another. 'Hide', 'Long', and 'Mid' correspond to increasingly risk-averse agents while 'Fast' corresponds to a risk-neutral agent. The following are the parameters we used: 'Fast', $\rho_+ = \rho_- = c_+ = c_- = 1.0$; 'Mid', $c_- = c_+ = 2.75$ and $\rho_+ = \rho_- = 0.35$; 'Long', $c_+ = c_- = 3.0$ and $\rho_+ = \rho_- = 0.33$; 'Hide', $c_+ = c_- = 4$ and $\rho_+ = \rho_- = 0.25$.

Agent Type	Prospect		Logarithmic		Entropic	
	Mean	Var	Mean	Variance	Mean	Variance
Fast	0.033	0.029	4.5e-3	5.1e-3	3.0e-4	6.0e-4
Mid	0.026	0.020	0.011	6.5e-3	9.3e-3	7.3e-3
Long	0.076	0.051	0.014	8.3e-3	1.7e-3	1.1e-3
Hide	0.144	0.011	0.050	0.043	9.0e-4	7.0e-4

Table 1: Error mean and variance, as measured by the TV distance, between the learned and true behaviors for various degrees of risk-aversion in agents with different types of value functions. The 'Fast' agent is risk-neutral; and 'Mid', 'Long', and 'Hide' have increasing degree of risk-aversion. For the prospect value function agents, the behaviors were generated by setting $c_+ = 1$, $\rho_- = \rho_+ = 0.5$ and varying c_- . The parameter values for the log value function agents are described in the caption of Figure 4. For the entropic map value function agents, we varied λ from 0.1 to -0.3 to generate the four behaviors.

general classes of behaviors each having different risk/reward tradeoffs. Each path in Fig. 4 presents a different risk profile, with the 'Fast' path being the riskiest but potentially the most rewarding, and the 'Hide' path being the most conservative.

We trained *true* agents with the prospect, entropic, and log value functions as described in Section 2.2, with various sets of parameters to generate the four behaviors. We then collected 10,000 sample trajectories from these agents, and learned the parameters of the value functions using our gradient-based approach. We calculated the distance in the TV norm defined in (25) between the policy in

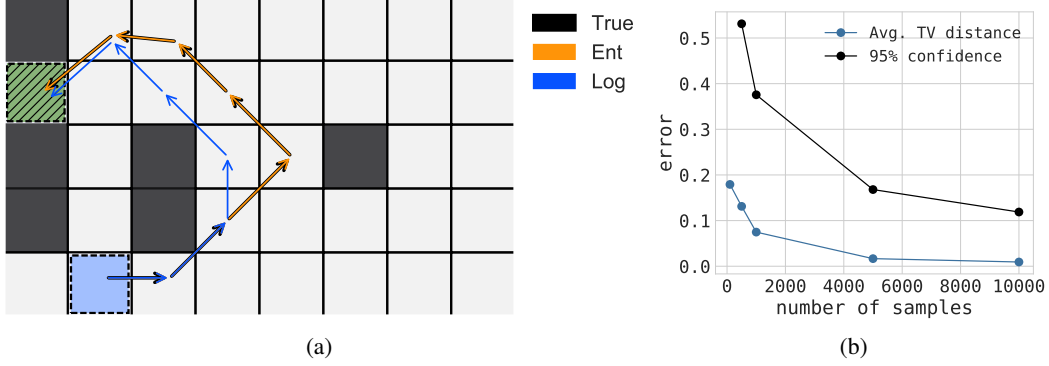


Figure 5: (a) Maximum Likelihood paths extracted from the optimal policies for agent’s with a prospect value function learned with an entropic and logarithm value function. The true and learned paths align exactly with one another for the true prospect and learned entropic agents. The learned logarithm agent, on the other hand, over approximates how risk-averse the true prospect agent is. The true prospect agent has parameters $(c_+, c_-, \rho_+, \rho_-) = (10, 1, 0.5, 0.5)$. The learned logarithm agent has parameters $(c_+, c_-, \rho_+, \rho_-) = (4.6, 0.3, 1, 1.5)$ and the learned entropic agent has $\lambda = -0.13$; (b) The sample complexity is of particular import for problems with humans in the loop due to the fact that there is often limited data. The depicted plot shows, for an agent with an entropic value function, the error between the empirical policy and the learned policy as a function of the number of samples.

state x of the true agent and the policy in state x of the learned agent. In Table 1 we report the mean distance across states, as well as the variance in the distance across states.

We first note that in all the cases considered in Table 1, the learned value functions produce policies that correctly match the maximum likelihood path of the true agent. We also note that performance for learning a prospect value function was markedly worse than learning entropic map or log value functions. This is most likely due to the fact that the prospect value function is not Lipschitz around the reference point. Thus, we have no guarantees of differentiability of Q^* with respect to θ , and the results are of worse quality. The entropic value function performs best of the three value functions, mainly because there is only one parameter to learn, and the rewards and losses are all relatively small.

We then learned parameters for agents operating under the entropic map and log value functions from data collected from an agent operating under the prospect value function and exhibiting the ‘Mid’ behavior. The maximum likelihood path of the true prospect agent and the learned entropic map and log value function agents is shown in Fig. 5a. The mean TV distance between the true prospect agent and the learned agents is shown in Table 2.

Of note with the experiments on learning prospect agents with logarithmic and entropic agents is the fact that, though the log value function fails to capture the correct maximum likelihood path, it is closer, on average to the true policy than the entropic value function which does learn the correct path. Thus, the logarithmic value function may generalize better, since it has learned the policy in general, better than the entropic value function. Further, we note that the logarithmic value function matches the true policy much better than the prospect value function. This is most likely due to the fact that the logarithmic value function approximates the prospect value function, but is globally Lipschitz. Thus, we have stronger guarantees of convergence.

Agent Type	Mean	Variance
Prospect	0.026	0.019
Entropic Map	0.095	4.3e-3
Logarithmic	0.021	1.0e-4

Table 2: Error mean and variance, as measured by the TV distance, between learned and true behavior for an true prospect agent learned by prospect, entropic, and logarithmic agents.

4.2 A Passenger’s View of Ride-Sharing

Reference dependence models are increasingly being used to model travel choices and activity scheduling [30]. More broadly, behavioral modeling has been used quite extensively in transportation to model travel choices (see, *e.g.*, [31–34]). Ride-sharing is a disruptive technology that offers commuters an alternative mode of transport.

Ride-sharing is a disruptive technology that offers commuters an alternative mode of transport. Many ride-sharing companies set prices based on both supply of drivers and demand of passengers. As a result, the price can fluctuate over time and space and passengers react differently to price changes depending on their risk preferences. We derive a toy model based on pricing data collected from the Uber API and travel time data collected via Uber Movement². The pricing data was collected at 3-minute intervals across 276 locations in Washington, D.C. (specified by Census blocks) from 2016 November 14–28³.

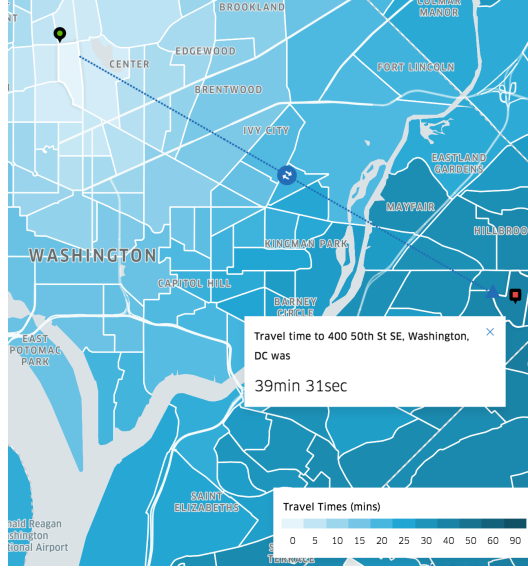


Figure 6: Origin and destination used in the ride-sharing example. The green marker is the origin and the red marker is the destination. The blue shaded scale shows the average travel time to locations starting from the green pin (origin). This graphic was obtained from the Uber Movement platform.

We examined several locations and hours which had different characteristics in terms of travel time and price statistics. However, the core risk-sensitive behaviors were the same. We generated a ride-sharing MDP for origin GPS= (−77.027046, 38.926749) and destination GPS= (−76.935773, 38.885964)⁴ in Washington D.C. at 5AM. Figure 6 shows the origin and destination on a map.

From the passenger’s view point, we model the ride-sharing MDP as follows. The action space is $A = \{0, 1\}$ where 0 corresponds to ‘wait’ and 1 corresponds to ‘ride.’ The state space $X = \mathcal{X} \times \mathcal{T} \cup \{x_f\}$ where $\mathcal{X} = \{1.0, 1.4, 1.8, 2.2\}$ is the part of the state corresponding to the price multiplier, $\mathcal{T} = \{0, \dots, T_f\}$ is the part of the state corresponding to the time index, and x_f is a terminal state representing the completed ride that occurs when a ride is taken. At time t , the state is notationally given by (x_t, t) . The reward r_t is modeled as a random variable that depends on the current price as well as a random variable $Z(t)$ for travel time whose distribution is estimated from the Uber Movement travel time data. In particular, for any time $t < T_f$ the reward is given by

$$R(x_{t+1}, a_t, x_t) = \begin{cases} \bar{R}, & a_t = 0 \text{ ('wait')} \\ S_t - x_t(p_{\text{base}} + p_{\text{mile}}D + p_{\text{min}}Z(t)), & a_t = 1 \text{ ('ride')} \end{cases} \quad (29)$$

²Uber Movement: <https://movement.uber.com/cities>

³The data was originally collected by and has been made publicly available here: <https://github.com/comp-journalism/2016-03-wapo-uber>

⁴Note that these correspond to Uber Movement id’s 197 and 113, respectively.

with $\bar{R} < 0$ a constant and where D is the distance in miles, S_t is a time dependent satisfaction (we selected it to linearly decrease in time from some initial satisfaction level), and p_{base} , p_{mile} , and p_{min} are the base, per mile, and per min prices, respectively. These prices are chosen based on Washington, DC Uber prices⁵. At the final time T_{final} , the agent is forced to take the ride if they have not selected to take a ride at a prior time. This reflects the fact that the agent presumably needs to get from their origin to their destination and the reward structure reflects the dissatisfaction the agent feels as a result of having to ultimately take the ride despite the potential desire to wait.

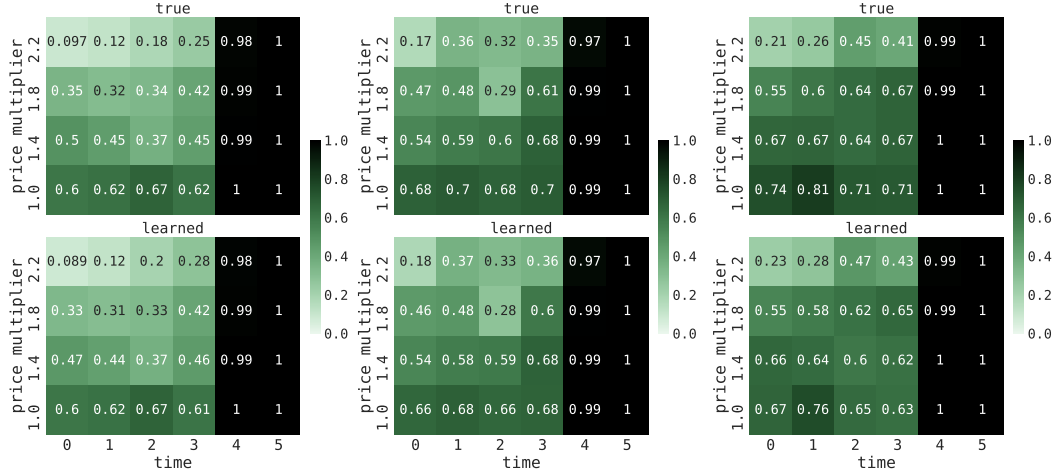
The transition probability kernel $P : X \times A \times X \rightarrow [0, 1]$ is estimated from the ride-sharing data. The travel-time data we have is available on an hourly basis. Hence, we use the 3 min price change data for each hour to derive a static transition matrix by empirically estimating the transition probabilities where we bin prices in the following way. For prices in $[1.0, 1.2)$, $x = 1.0$; for prices in $[1.2, 1.6)$, $x = 1.4$; for prices in $[1.6, 2.0)$, $x = 1.8$; otherwise $x = 2.2$. In the time periods we examined, the max price multiplier was 2.0. We selected the reference point to be $\bar{y} = 0$ and the acceptance level to be $\bar{r} = 0$.

The estimated transition matrix for the location we selected is given by

$$P = \begin{bmatrix} 0.876 & 0.099 & 0.017 & 0.008 \\ 0.347 & 0.412 & 0.167 & 0.074 \\ 0.106 & 0.353 & 0.259 & 0.282 \\ 0.086 & 0.219 & 0.143 & 0.552 \end{bmatrix} \quad (30)$$

for each time. The travel time distribution is a standard normal distribution truncated to the upper and lower bounds specified by the Uber Movement data. Measured in seconds, we use location parameter 2371, scale parameter 100, and 1554 and 3619 as the upper and lower bound, respectively.

Figures 7, 8, and 9 show the state space as a grid with the probability of taking a ride under the true and learned optimal policies overlaid on each state. For the examples in Figure 7 and Figure 8, we vary the degree of risk-sensitivity in gains and losses, respectively while holding the other at risk-neutral. For the example in Figure 9, on the other hand, we vary both the risk-sensitivity of gains and losses simultaneously. Note that there is strong experimental evidence supporting the fact that human decision-makers tend to be risk-averse in gains and risk-seeking in losses. Figure 9c is an example of risk preferences aligned with this type of behavior.



(a) risk-averse in gains, $\rho_+ = 0.5$ (b) risk-neutral, $\rho_+ = \rho_- = 1.0$ (c) risk-seeking in gains, $\rho_+ = 1.5$

Figure 7: Prospect Agents with $c_+ = c_- = 1.0$ and $\rho_- = 1.0$. The plots show the probabilities of taking a ride in each state under the true and learned optimal policies. In light of the reward structure shown in Figure 10a, we see that when $x_t \in \{1.0, 1.4\}$ the reward is positive on average. The more risk-averse an agent is in gains, the less likely they are to take a ride.

⁵The base, per min, and per mile prices can be found here: <http://uberestimate.com/prices/Washington-DC/>

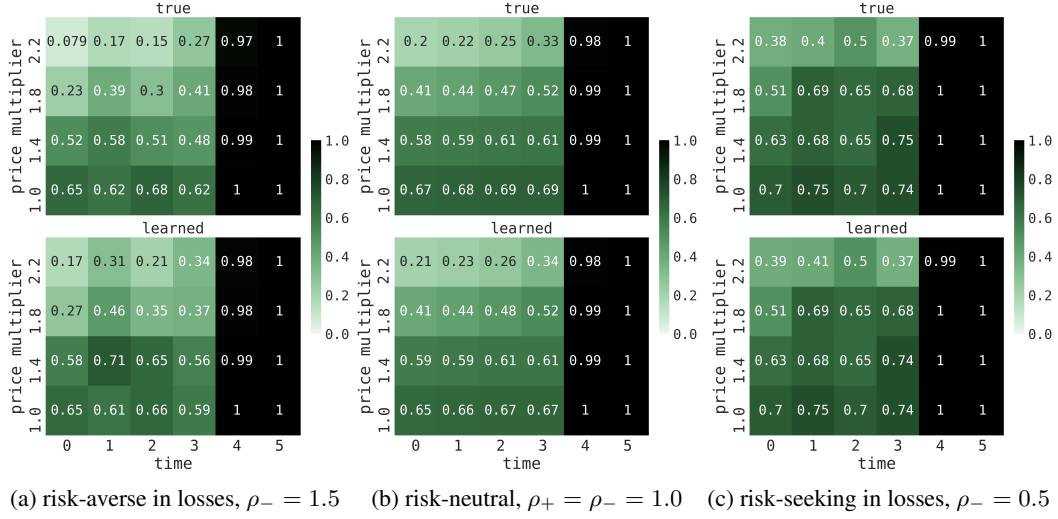


Figure 8: Prospect Agents with $c_+ = c_- = 1.0$ and $\rho_+ = 1.0$. The plots show the probabilities of taking a ride in each state under the true and learned optimal policies. In light of the reward structure shown in Figure 10a, we see that when $x_t \in \{1.8, 2.2\}$ the reward is negative on average. The more risk-seeking an agent is in losses, the more likely they are to take a ride.

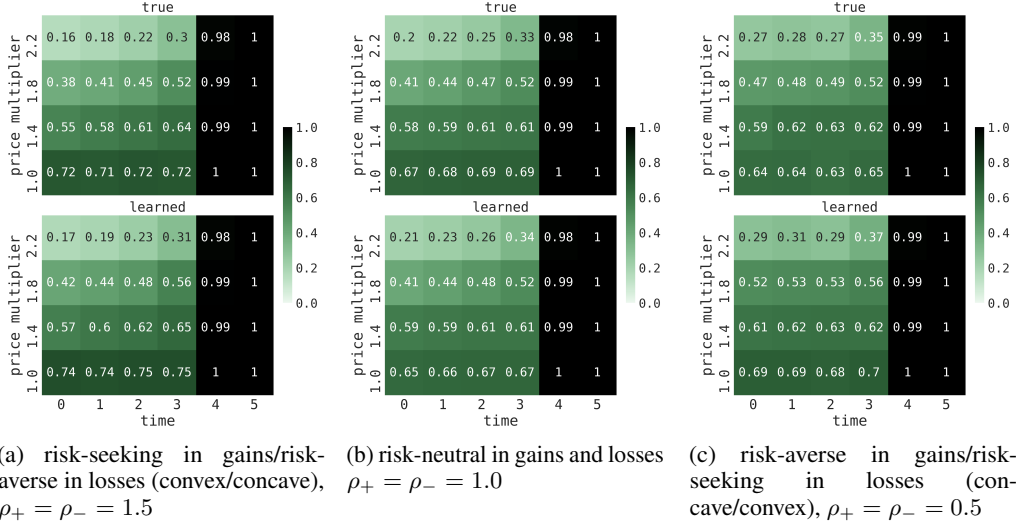


Figure 9: Prospect agents with $c_+ = c_- = 1.0$. The plots show the probabilities of taking a ride in each state under the true and learned optimal policies. The value function used for the right most graphic (Figure 9c) is most representative of human decision-making since humans tend to be risk-averse in gains and risk-seeking in losses. We see the same trend as before: the more risk-averse, the less likely to take the ride.

In Figure 10a we show a realization of the reward r_t for different time instances. It turns out that in expectation, the reward is positive for $x_t \in \{1.0, 1.4\}$ and is negative for $x_t \in \{1.8, 2.2\}$. Across the examples depicted in Figures 7, 8, and 9, we see the same trend: the more risk-averse (in gains or losses), the less likely the passenger is to take the ride.

In Table 3, we show the mean and variance of the total variation error for the ride-sharing example where we varied the risk-sensitivity of gains (i.e. the example depicted in Figure 7) using both agents with prospect and entropic value functions. The errors for the other two examples are similar in size. The entropic map performs slightly better this is largely due to the fact that λ is the only

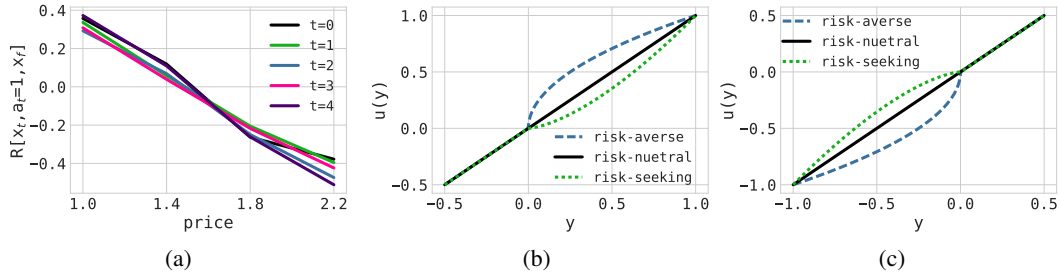


Figure 10: (a) Rewards for the ride-sharing example. Notice that the rewards can either be gains (positive values) or losses (negative values) given that we take the reference point to be $\bar{y} = 0$. (b) Prospect value functions corresponding to the example in Figure 7 where the degree of risk-sensitivity in gains is explored. (c) Prospect value functions corresponding to the example in Figure 8 where the degree of risk-sensitivity in losses is explored.

parameter to learn. Note that the entropic agent performs worse compared to itself when learning risk-neutral preferences which occurs near $\lambda = 0$ while the prospect agent has its worse performance on risk-seeking preferences.

<i>Agent Type</i> Risk Preference	<i>Prospect</i>		<i>Entropic</i>		<i>Prospect/Entropic</i>	
	Mean	Variance	Mean	Variance	Mean	Variance
risk-averse in gains	0.023	5.3e-4	2.7e-4	5.3e-8	0.021	5.0e-4
risk-neutral	0.006	4.9e-5	1.7e-3	2.3e-6	1.2e-3	1.2e-6
risk-seeking in gains	0.006	8.1e-5	7.2e-4	4.9e-7	0.011	1.4e-4

Table 3: Error computed using the total variation for the example where we vary the degree of risk-sensitivity in gains (i.e. the example shown in Figure 7). We report the average error and the variance between the policies. The last column shows the error when using an entropic agent to learn a prospect agent.

5 Discussion

We present a new gradient based technique for learning risk-sensitive decision making models of humans amidst automation. We find that while there are a number of technical issues related to learning prospect theory based agents—namely, their value functions are not Lipschitz for parameter combinations that best capture human decision making (i.e. when $0 < \rho_+, \rho_- < 1$)—we are able to still numerically learn the policies of these agents. Moreover, we introduce a logarithm-based value function that well-approximates the convex-concave structure of the prospect theory values while satisfying the assumptions of our theorems. We demonstrated the algorithms performance for agents based on several types of behavioral models and do so on two examples: the canonical Grid World problem and a passenger’s view ride-sharing where the parameters of the ride-sharing MDP were learned from real-world data.

Looking forward, we remark that we assumed the reference point was known and fixed. We are examining techniques that have formal performance guarantees (e.g., on convergence) that allow us to simultaneously estimate reference points. We are also examining the use of other risk-metrics that will allow us to leverage the benefits of cumulative prospect theory which has been shown to be more flexible in approximating human decision-making. As a final remark, we previously remarked that the complexity bound we derive, since it is on the per-state level, suggests ways of designing data collection schemes to better understand the agent’s actions in less explored regions of the state space. We currently exploring how such information can be used to adaptively produce the most informative demonstrations from human agent’s. This is challenging since some of the options in the state-action space may result in a very high risk to the agent and thus, inducing such state-action pairs may be viewed unfavorably by the agent.

A Proof of Theorem 2

Let us briefly remark that the gradient algorithm and Theorem 2 are consistent with the gradient descent framework which uses the *contravariant* gradient for learning as introduced in [35] for Riemannian parameter spaces Θ . Of course, when Θ is Euclidean and the coordinate system is orthonormal, the gradient we normally use (*covariant* derivative) coincides with the contravariant gradient. However, this suggests our techniques will not generalize to admissible parameter spaces with more structure. Moreover, as is pointed out in [23], the trajectories that result from the solution to the gradient algorithm are equivalent up to reparameterization through a smooth invertible mapping with a smooth inverse. Contravariant gradient methods have been shown to be asymptotically efficient in a probabilistic sense and thus, they tend to avoid *plateaus* [35, 36].

Before we dive into the proof of Theorem 2, let us introduce some definitions and useful propositions.

Definition 2 (Fréchet Subdifferentials) *Let U be a Banach space and U^* its dual. The Fréchet subdifferential of $f : U \rightarrow \mathbb{R}$ at $u \in U$, denoted by $\partial f(u)$ is the set of $u^* \in U^*$ such that*

$$\liminf_{h \rightarrow 0, h \neq 0} \|h\|^{-1} (f(u+h) - f(u) - \langle u^*, h \rangle) \geq 0 \quad (31)$$

We require two propositions to complete the proof.

Proposition 1 ([23, 37]) *For a finite family $(f_i)_{i \in I}$ of real-valued functions (where I is a finite index set) defined on U , let $f(u) = \max_{i \in I} f_i(u)$. If $u^* \in \partial f_i(u)$ and $f_i(u) = f(u)$, then $u^* \in \partial f(u)$. If $f_1, f_2 : U \rightarrow \mathbb{R}$, $\alpha_1, \alpha_2 \geq 0$, then $\alpha_1 \partial f_1 + \alpha_2 \partial f_2 \subset \partial(\alpha_1 f_1 + \alpha_2 f_2)$.*

Proposition 2 ([23, 38]) *Suppose that $(f_n)_{n \in \mathbb{N}}$ is a sequence of real-valued functions on U which converge pointwise to f . Let $u \in U$, $u_n^* \in \partial f_n(u) \subset U^*$ and suppose that (u_n^*) is weak*-convergent to u^* and is bounded. Moreover, suppose that at u , for any $\varepsilon > 0$, there exists an $N > 0$ and $\delta > 0$ such that for any $n \geq N$, $h \in B_U(0, \delta)$ ⁶, $f_n(u+h) \geq f_n(u) + \langle u_n^*, h \rangle - \varepsilon \|h\|$. Then $u^* \in \partial f(u)$.*

Proof of Theorem 2.

Let $Q_0(x, a, \theta) \equiv 0$. Then it is trivial that $Q_0(x, a, \theta)$ is locally Lipschitz in θ on Θ . Supposing that $Q_t(x, a, \theta)$ is L_t -locally Lipschitz in θ , then we need to show that $TQ_t(x, a, \theta)$ is locally Lipschitz.

Since $\tilde{u} \equiv u - \bar{r}$, it also satisfies Assumption 1. Let $L_y = \min\{L_y(\theta) | \theta \in \Theta\}$. and define $v(x, \theta) = \max_{a'} Q(x, a', \theta)$. Then we have that

$$\begin{aligned} \|TQ_t(\theta) - TQ_t(\theta')\| &\leq \alpha L_y \|y(\theta) - y(\theta')\| + \alpha L_\theta \|\theta - \theta'\| + \|Q_t(\theta) - Q_t(\theta')\| \\ &\leq \alpha \gamma L_y \|v(s', \theta) - v(s', \theta')\| + \alpha L_\theta \|\theta - \theta'\| + (1 - \alpha L_y) \|Q_t(\theta) - Q_t(\theta')\| \\ &\leq (1 - \alpha(1 - \gamma)L_y) \|Q_t(\theta) - Q_t(\theta')\| + \alpha L_\theta \|\theta - \theta'\| \\ &\leq ((1 - \alpha(1 - \gamma)L_y)L_t + \alpha L_\theta) \|\theta - \theta'\| \end{aligned}$$

where we have suppressed the dependence of $Q_t(\cdot, \cdot, \theta)$ on (x, a) . Hence, letting $\bar{\alpha} = 1 - \alpha(1 - \gamma)L_y$ we have that $TQ_t(\cdot, \cdot, \theta)$ is L_{t+1} -locally Lipschitz with $L_{t+1} = \bar{\alpha}L_t + \alpha L_\theta$. With $L_0 = 0$, by iterating we get that

$$L_{t+1} = (\bar{\alpha}^t + \dots + \bar{\alpha} + 1)\alpha L_\theta$$

As stated in Section 3.2, T is a contraction so that $T^n Q_0 \rightarrow Q_\theta^* = Q^*(\cdot, \cdot, \theta)$ as $n \rightarrow \infty$. Hence, by the above argument Q_θ^* is $\alpha L_\theta / (1 - \bar{\alpha})$ -Lipschitz continuous.

Now, consider a fixed vector $\theta \in \mathbb{R}^d$. We now show that the operator S acting on the space of function $\phi_\theta : X \times A \rightarrow \mathbb{R}^d$ defined by

$$\begin{aligned} (S\phi_\theta)(x, a) &= \alpha \mathbb{E}^\mu \left[D_\theta \tilde{u}(y(\theta), \theta) + D_y \tilde{u}(y(\theta), \theta) \left(\gamma \sum_{a' \in A} \pi(a'|x) \phi_\theta(x', a') - \phi_\theta(x, a) \right) \right] \\ &\quad + \phi_\theta(x, a) \end{aligned} \quad (32)$$

⁶ $B_U(0, \delta) \subset U$ is a δ -ball around 0.

is a contraction where π is any greedy policy with respect to $Q_\theta = Q(\cdot, \cdot, \theta)$. Indeed,

$$\begin{aligned} (S\phi_\theta - S\phi'_\theta)(x, a) &= \alpha \mathbb{E}^\mu \left[D_y \tilde{u}(y(\theta), \theta) \cdot \left(\gamma \sum_{a' \in A} \pi(a'|x') (\phi_\theta(x', a') - \phi'_\theta(x', a')) \right. \right. \\ &\quad \left. \left. - (\phi_\theta(x, a) - \phi'_\theta(x, a)) \right) \right] + \phi_\theta(x, a) - \phi'_\theta(x, a) \\ &\leq \alpha \gamma \mathbb{E}^\mu \left[D_y \tilde{u}(y(\theta), \theta) \cdot \left(\sum_{a' \in A} \pi(a'|x') (\phi_\theta(x', a') - \phi'_\theta(x', a')) \right) \right] \\ &\quad + \left(1 - \alpha \mathbb{E}^\mu [D_y \tilde{u}(y(\theta), \theta)] \right) (\phi_\theta(x, a) - \phi'_\theta(x, a)) \end{aligned}$$

so that $\|(S\phi_\theta - S\phi'_\theta)(x, a)\| \leq (1 - \alpha(1 - \gamma)L_y)\|\phi_\theta - \phi'_\theta\|_\infty$ since for a differentiable Lipschitz map the norm of the derivative is always bounded above by the Lipschitz constant. Thus, $\bar{\alpha}$ is the required constant for ensuring S is a contraction. We remark that S operates on each of the d components of θ separately and hence, it is a contraction when restricted to each individual component.

Let π denote a greedy policy with respect to Q_θ^* and let π_n be a sequence of policies that are greedy with respect to $Q_n = T^n Q_0$ where ties are broken so that $\sum_{(x,a) \in X \times A} |\pi(a|x) - \pi_n(a|x)|$ is minimized. Then for large enough n , $\pi_n = \pi$. Denote by S_{π_n} the map S defined in (32) where π_n is the implemented policy. Consider the sequence $\phi_{\theta,n}$ such that $\phi_{\theta,0} = 0$ and $\phi_{\theta,n+1} = S_{\pi_n} \phi_{\theta,n}$. For large enough n , $\phi_{\theta,n+1} = S_\pi \phi_{\theta,n}$. Applying the (local) contraction mapping theorem (see, e.g., [39, Theorem 3.18]) we get that $\lim_{n \rightarrow \infty} S^n \phi_0$ converges to a unique fixed point. Moreover, by induction and Proposition 1, $\phi_{\theta,n}(x, a) \in \partial_\theta Q_n(x, a, \theta)$. Hence, by Proposition 2, the limit is a subdifferential of Q_θ^* since \tilde{u} is Lipschitz on Y and θ , the derivatives of \tilde{u} are uniformly bounded. Since by part (a) Q_θ^* is locally Lipschitz in θ , Rademacher's Theorem (see, e.g., [27, Thm. 3.1]) tells us it is differentiable almost everywhere (except a set of Lebesgue measure zero). Since Q_θ^* is differentiable, its subdifferential is its derivative. ■

References

- [1] A. Tversky and D. Kahneman, "Rational choice and the framing of decisions," *The Journal of Business*, vol. 59, no. 4, pp. S251–S278, 1986.
- [2] —, "Loss aversion in riskless choice: A reference-dependent model," *Quarterly J. Economics*, vol. 106, no. 4, pp. 1039–1061, 1991.
- [3] Y. Shen, W. Stannat, and K. Obermayer, "Risk-Sensitive Markov Control Processes," *SIAM J. Control Optimization*, vol. 51, no. 5, pp. 3652–3672, 2013.
- [4] S. I. Marcus, E. Fernández-Gaucherand, D. Hernández-Hernández, S. Coraluppi, and P. Fard, *Risk Sensitive Markov Decision Processes*. Birkhäuser Boston, 1997, pp. 263–279.
- [5] P. Geibel and F. Wysotzki, "Risk-sensitive reinforcement learning applied to control under constraints," *Journal of Artificial Intelligence Research*, vol. 24, pp. 81–108, 2005.
- [6] P. L.A., C. Jie, M. Fu, S. Marcus, and C. Szepesvári, "Cumulative prospect theory meets reinforcement learning: Prediction and control," in *Proc. 33rd Intern. Conf. on Machine Learning*, vol. 48, 2016.
- [7] Y. Shen, M. J. Tobia, and K. Obermayer, "Risk-sensitive reinforcement learning," *Neural Computation*, vol. 26, pp. 1298–1328, 2014.
- [8] O. Mihatsch and R. Neuneier, "Risk-sensitive reinforcement learning," *Machine Learning*, vol. 49, no. 2, pp. 267–290, 2002.
- [9] A. J. Nagengast, D. A. Braun, and D. M. Wolpert, "Risk-sensitive optimal feedback control accounts for sensorimotor behavior under uncertainty," *PLOS Computational Biology*, vol. 6, no. 7, pp. 1–15, 2010.
- [10] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–291, 1979.

- [11] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *Journal of Risk Uncertainty*, vol. 5, no. 4, pp. 297–323, Oct 1992.
- [12] H. Simon, "Bounded rationality in social science: Today and tomorrow," *Mind & Society*, vol. 1, no. 1, pp. 25–39, Mar. 2000.
- [13] A. Tversky and D. Kahneman, "The framing of decisions and the psychology of choice," *Science*, vol. 211, no. 4481, pp. 453–458, Jan. 1981.
- [14] C. F. Camerer, "An experimental test of several generalized utility theories," *J. Risk and Uncertainty*, vol. 2, no. 1, pp. 61–104, 1989.
- [15] R. Gonzalez and G. Wu, "On the shape of the probability weighting function," *Cognitive Psychology*, vol. 38, no. 1, pp. 129–166, 1999.
- [16] G. Wu and R. Gonzalez, "Curvature of the probability weighting function," *Management Science*, vol. 42, no. 12, pp. 1676–1690, 1996.
- [17] H. Föllmer and A. Schied, "Convex measures of risk and trading constraints," *Finance and Stochastics*, vol. 6, no. 4, pp. 429–447, 2002.
- [18] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.
- [19] H. Föllmer and A. Schied, *Stochastic Finance: An Introduction in Discrete Time*. Walter de Gruyter, 2004.
- [20] R. Cavazos-Cadena, "Optimality equations and inequalities in a class of risk-sensitive average cost markov decision chains," *Mathematical Methods of Operations Research*, vol. 71, no. 1, pp. 47–84, 2010.
- [21] S. P. Coraluppi and S. I. Marcus, "Mixed risk-neutral/minimax control of discrete-time, finite-state markov decision processes," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 528–532, 2000.
- [22] H. Robbins and D. Siegmund, *A Convergence Theorem for Non Negative Almost Supermartingales and Some Applications*. Springer New York, 1985, pp. 111–135.
- [23] G. Neu and C. Szepesvári, "Apprenticeship learning using inverse reinforcement learning and gradient methods," in *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, 2007, pp. 295–302.
- [24] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.
- [25] J. N. Tsitsiklis, "Asynchronous stochastic approximation and q-learning," *Machine Learning*, vol. 16, no. 3, pp. 185–202, 1994.
- [26] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.
- [27] J. Heinonen, "Lectures on lipschitz analysis," *Lectures at the 14th Jyväskylä Summer School*, 2004.
- [28] P. W. Millar, "Asymptotic minimax theorems for the sample distribution function," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 48, no. 3, pp. 233–252, 1979.
- [29] P. Massart, "The tight constant in the dvoretzky-kiefer-wolfowitz inequality," *The Annals of Probability*, vol. 18, pp. 1269–1283, 1990.
- [30] Q. Li, F. Liao, H. J. P. Timmermans, and J. Zhou, "A reference-dependent user equilibrium model for activity-travel scheduling," *Transportation*, vol. 43, no. 6, pp. 1061–1077, 2016.
- [31] P. Burnett, "Disaggregate behavioral models of travel decisions other than mode choice: a review and contribution to spatial choice theory," *Transportation Research Board Special Report*, vol. 149, 1974.

- [32] F. S. Koppelman, “Non-linear utility functions in models of travel choice behavior,” *Transportation*, vol. 10, no. 2, pp. 127–146, 1981.
- [33] E. Avineri and P. Bovy, “Identification of parameters for a prospect theory model for travel choice analysis,” *Transportation Research Record*, vol. 2082, 2008.
- [34] H. Xu, J. Zhou, and W. Xu, “A decision-making rule for modeling travelers’ route choice behavior based on cumulative prospect theory,” *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 2, pp. 218–228, 2011.
- [35] S.-I. Amari, “Natural gradient works efficiently in learning,” *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.
- [36] J. Peters, S. Vijayakumar, and S. Schaal, “Natural actor-critic,” in *Proceedings of the 16th European Conference on Machine Learning*, 2005, pp. 280–291.
- [37] A. Y. Kruger, “On fréchet subdifferentials,” *J. Mathematical Sciences*, vol. 116, no. 3, 2003.
- [38] J. Penot, “On the interchange of subdifferentiation and epi-convergence,” *J. Mathematical Analysis and Applications*, vol. 196, no. 2, pp. 676–698, 1995.
- [39] S. S. Sastry, *Nonlinear Systems*. Springer, 1999.